
An Introduction to PatPir

Raphaël Jauslin

14 décembre 2018

TABLE DES MATIÈRES

1	Overview	2
2	Pre-treatment	2
2.1	Merging	2
2.2	Demultiplexing	3
2.2.1	Simple tag	3
2.2.2	Double tag	3
2.3	Quality check	4
2.4	Remove 'N' fasta	4
3	Database	4
3.1	PR2	4
3.2	Other base	4
4	Dereplication	4

1 OVERVIEW

This package is a tool to facilitate the pre-treatment and the treatment of NGS data. The tools are implemented to work on fastq and fasta file. This introduction will, step-by-step, explain how to use the package and which functions you should use in order to obtain your data merged, demultiplexed and cleaned. This will suppose that you have two .fastq R1 and R2 and a barcode file that contains the information for the demultiplexing step. Firstly, we need to download and install the package PatPir. You could find the package in the github repository : <https://github.com/Rjauslin/PatPir>. You should launch the following commands in R or Rstudio in order to install PatPir.

```
install.packages("devtools")
devtools::install_github("Rjauslin/PatPir@master")
```

2 PRE-TREATMENT

This is the first step of the pipeline. The inputs are the original files from the server. They are probably named something like xxx_R1.fastq and xxx_R2.fastq. We will explain how the program deal with the main three steps, merging, demultiplexing, and cleaning. During all the process we will suppose that you have put only the two fastq files and the information needed for the demultiplexing (see Section 2.2).

```
library(PatPir)

#Linux
pathFolder <- "/home/raphael/Documents/...../working_directory/"
#Windows
pathFolderWindows <- "C:/Users/raphael/...../working_directory/"
```

All the pre-treatment is wrapped inside a function that call the different programs. So you only have to check the parameter of the main function `preTreatment`.

```
preTreatment(pathFolder,
  m = 10, # min overlap
  M = 100, # max overlap
  x = 0.25, # max mismatch density
  t = 4, # number of threads
  mismatch = FALSE, # allows 1 mismatch in tag if TRUE
  err = 0.01, # expected error
  slide = 50, # sliding window
  minlength = 60) # minimum length of sequences considered
```

2.1 MERGING

The merging step currently implemented is done by the program FLASH [Magoč and Salzberg, 2011]. We have allowed some possible change.

- m The minimum required overlap length between two reads to provide a confident overlap.
- M Maximum overlap length expected in approximately 90% of read pairs.
- x Maximum allowed ratio between the number of mismatched base pairs and the overlap length.
- t Set the number of worker threads.

2.2 DEMULTIPLEXING

We will here define the requirements for the demultiplexing part.

2.2.1 SIMPLE TAG

In case of simple barcode you should only put one additional file called `barcode.txt`. The demultiplexing step is implemented by the program `PatPil` that is hidden in the package. The function calls `D_simple_tag` that could be used from the shell by the following command. So it is really important that the barcode file have the right format.

```
./PatPil D_simple_tag -f ./merged.fastq -o ./outputFolder/ -b ./barcodes.txt -mismatch
```

The only thing that you should care is that your `barcode.txt` file is of the following form. **Specifically, it is really important to add the .fastq and not another extension file and you should verify that the separator between the tags and the names of the files is a tab.**

```
ACGAGTGCCT 01.fastq
ACGCTCGACA 02.fastq
AGACGCACTC 03.fastq
AGCACTGTAG 04.fastq
ATCAGACACG 05.fastq
ATATCGCGAG 06.fastq
CGTGTCTCTA 07.fastq
CTCGCGTGTC 08.fastq
...
```

If your barcode file follow all these requirements, the demultiplexing part should work properly.

2.2.2 DOUBLE TAG

In case of double barcode you should only put three additional files called `forwardtag.txt`, `reversetag.txt` and `primer.txt`. The `forwardtag` and `reversetag` contains the barcodes that it supposed to be at the beginning and at the end of your sequences. **You should not include some extensions file such as .fastq or .fq in these two files.**

```
ACACACAC ForwardTag1
ACGACTCT ForwardTag2
ACGCTAGT ForwardTag3
ACTATCAT ForwardTag4
...
```

```
ACACACAC ReverseTag1
ACGACTCT ReverseTag2
ACGCTAGT ReverseTag3
ACTATCAT ReverseTag4
...
```

The `primer.txt` should contain only two information, that forward primer and the reverse primer **in this order**. The primers could have some uncertain nucleotide. The program transform and do all the combination by creating two new files called `primerforward` and `primerreverse`. The transformations are done with the following table.

```
CAAAATCATAAAGATATTGGDAC GAAATTTCCDGGDTATMGAATGG
```

```
R = AG
Y = CT
S = GC
W = AT
K = GT
M = AC
B = CGT
D = AGT
H = ACT
V = ACG
```

If you named your file with the good title, with the right format, the demultiplexing part should work properly.

2.3 QUALITY CHECK

The quality check currently implemented evaluating the expected error in a 50 bp sliding window and discarding sequences with more than 1% of error in the worst quality window [De Vargas et al., 2015].

2.4 REMOVE 'N' FASTA

3 DATABASE

3.1 PR2

3.2 OTHER BASE

4 DEREPLICATION

RÉFÉRENCES

- [De Vargas et al., 2015] De Vargas, C., Audic, S., Henry, N., Decelle, J., Mahé, F., Logares, R., Lara, E., Berney, C., Le Bescot, N., Probert, I., Carmichael, M., Poulain, J., Romac, S., Colin, S., Aury, J. M., Bittner, L., Chaffron, S., Dunthorn, M., Engelen, S., Flegontova, O., Guidi, L., Horák, A., Jaillon, O., Lima-Mendez, G., Lukeš, J., Malviya, S., Morard, R., Mulot, M., Scalco, E., Siano, R., Vincent, F., Zingone, A., Dimier, C., Picheral, M., Searson, S., Kandels-Lewis, S., Acinas, S. G., Bork, P., Bowler, C., Gorsky, G., Grimsley, N., Hingamp, P., Iudicone, D., Not, F., Ogata, H., Pesant, S., Raes, J., Sieracki, M. E., Speich, S., Stemmann, L., Sunagawa, S., Weissenbach, J., Wincker, P., Karsenti, E., Boss, E., Follows, M., Karp-Boss, L., Krzic, U., Reynaud, E. G., Sardet, C., Sullivan, M. B., and Velayoudon, D. (2015). Eukaryotic plankton diversity in the sunlit ocean. *Science*, 348(6237).
- [Magoč and Salzberg, 2011] Magoč, T. and Salzberg, S. L. (2011). FLASH : Fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*.